IN THE CLAIMS

1. (original) A method, comprising:

inserting an allocation instruction within a routine if a function call instruction is found within said routine.

2. (original) The method of claim 1 further comprising configuring said allocation instruction to allocate only for the live information that exists within said routine when said inserted allocation instruction is executed.

3. (original) The method of claim 2 wherein said live information is determined by identifying information that is referred to before and after said function call.

4. (original) The method of claim 3 wherein said information identified after said function call extends to an exit block of said routine.

5. (original) The method of claim 4 wherein the worst case path to said exit block is allocated for.

6. (original) The method of claim 3 wherein said information identified after said function call extends to a post-dominator block of said routine.

7. (original) The method of claim 6 wherein the worst case path to said post-dominator block is allocated for.

8. (original) The method of claim 2 wherein said live information is information that is local to said routine.

9. (currently amended) The method of claim 8 wherein [the]a processor said routine is to be executed upon has its associated register space partitioned into register space used only for local information and register space used only for global information, said allocation instruction pertaining only to said register space used for local information.

10. (original) The method of claim 2 wherein said live information includes global information.

11. (original) The method of claim 1 wherein said allocation instruction is inserted just before said function call.

12. (original) The method of claim 1 wherein said allocation instruction is inserted in a pre-dominator basic block of said function call.

13. (original) The method of claim 12 wherein said allocation instruction is inserted in said pre-dominator basic block of said function call if there exists a post-dominator basic block of said function call.

14. (currently amended) A method comprising:

inserting multiple allocation instructions within a routine by searching for one or more functional characteristics within said routine and inserting an allocation instruction within said routine [if a]for each functional characteristic that is discovered.

15. (currently amended) The method of claim 14 wherein said one or more functional characteristics include a loop that exists within a control flow graph of said routine[corresponds to a functional characteristic].

16. (currently amended) The method of claim 15 wherein [said]the allocation instruction inserted for said loop is inserted above said loop in said control flow graph.

17. (original) The method of claim 16 wherein said allocation instruction allocates for a worst case path to an exit block of said routine.

18. (original) The method of claim 16 wherein said allocation instruction allocates for a worst case path to a post-dominator block of said routine.

19. (currently amended) The method of claim 14 wherein said one or more functional characteristics include a software pipelined loop[corresponds to a functional characteristic].

20. (currently amended) The method of claim 19 wherein [said]the allocation instruction inserted for said software pipelined loop is inserted above said loop in said control flow graph.

21. (original) The method of claim 20 wherein said allocation instruction allocates for a worst case path to an exit block of said routine.

22. (original) The method of claim 21 wherein said allocation instruction allocates for a worst case path to a post-dominator block of said routine.

23. (currently amended) The method of claim 14 wherein said one or more functional characteristics include a function call[corresponds to a functional characteristic].

24. (original) The method of claim 14 further comprising determining the number of registers to be allocated for an allocation instruction after a functional characteristic is found.

25. (currently amended) The method of claim 24 wherein all of said one or more functional characteristics within said routine are discovered before said determining is performed.

26. (original) The method of claim 24 wherein said determining is performed before a next functional characteristic is discovered.

27. (original) The method of claim 14 further comprising building an understanding of said routine's control flow graph before said searching is performed.

28. (currently amended) A method, comprising:

a) performing a first allocation for a first amount of register space at the entry block of a routine;

b) performing a second allocation for a second amount of register space for the live information within said routine at the time of said second allocation;

c) performing a function call to a second routine; and,

d) performing a third allocation for a third amount of register space at the entry block of said second routine, said third amount of register space and said first amount register space having a common register.

29. (original) The method of claim 28 wherein said live information is determined by identifying information that is referred to before and after said function call.

30. (original) The method of claim 29 wherein said information identified after said function call extends to an exit block of said routine.

31. (original) The method of claim 30 wherein the worst case path to said exit block is allocated for.

32. (original) The method of claim 29 wherein said information identified after said function call extends to a post-dominator block of said routine.

33. (original) The method of claim 32 wherein the worst case path to said post-dominator block is allocated for.

34. (original) The method of claim 28 wherein said live information is information that is local to said routine.

35. (currently amended) The method of claim 34 wherein [the]a processor said routine is to be executed upon has its associated register space partitioned into register space used only for local information and register space used only

for global information, said allocation instruction pertaining only to said register space used for local information.

36. (original) The method of claim 28 wherein said live information includes global information.

37. (original) The method of claim 28 wherein said second allocation is performed just before said function call.

38. (original) The method of claim 28 wherein said second allocation is performed in a pre-dominator basic block of said function call.

39. (original) The method of claim 28 wherein said second allocation is performed in said pre-dominator basic block of said function call if there exists a post-dominator basic block of said function call.

40. (original) The method of claim 28 further comprising compiling said routine.

41. (original) A machine readable medium having stored thereon sequences of instructions which are executable by a digital processing system, and which, when executed by the digital processing system, cause the system to perform a method comprising:

inserting an allocation instruction within a routine if a function call instruction is found within said routine.

42. (currently amended) The machine readable medium of claim 41 further comprising [sequences of] instructions which cause [the system]a processor that executes said routine to configure said allocation instruction to allocate only for the live information that exists within said routine when said inserted allocation instruction is executed.

43. (original) The machine readable medium of claim 42 wherein said live information is determined by identifying information that is referred to before and after said function call.

44. (original) The machine readable medium of claim 43 wherein said information identified after said function call extends to an exit block of said routine.

45. (original) The machine readable medium of claim 44 wherein the worst case path to said exit block is allocated for.

46. (original) The machine readable medium of claim 43 wherein said information identified after said function call extends to a post-dominator block of said routine.

47. (original) The machine readable medium of claim 46 wherein the worst case path to said post-dominator block is allocated for.

48. (original) The machine readable medium of claim 42 wherein said live information is information that is local to said routine.

49. (currently amended) The machine readable medium of claim 48 wherein [the]said processor said routine is to be executed upon has its associated register space partitioned into register space used only for local information and register space used only for global information, said allocation instruction pertaining only to said register space used for local information.

50. (original) The machine readable medium of claim 42 wherein said live information includes global information.

51. (original) The machine readable medium of claim 41 wherein said allocation instruction is inserted just before said function call.

52. (original) The machine readable medium of claim 41 wherein said allocation instruction is inserted in a pre-dominator basic block of said function call.

53. (original) The machine readable medium of claim 52 wherein said allocation instruction is inserted in said pre-dominator basic block of said function call if there exists a post-dominator basic block of said function call.

54. (currently amended) A machine readable medium having stored thereon sequences of instructions which are executable by a digital processing system, and which, when executed by the digital processing system, cause the system to perform a method comprising:

inserting multiple allocation instructions within a routine by searching for one or more functional characteristics within said routine and inserting an

allocation instruction within said routine [if a]for each functional characteristic that is discovered.

55. (currently amended) The machine readable medium of claim 54 wherein said one or more functional characteristics include a loop that exists within a control flow graph of said routine[corresponds to a functional characteristic].

56. (currently amended) The machine readable medium of claim 55 wherein [said]the allocation instruction inserted for said loop is inserted above said loop in said control flow graph.

57. (original) The machine readable medium of claim 56 wherein said allocation instruction allocates for a worst case path to an exit block of said routine.

58. (original) The machine readable medium of claim 56 wherein said allocation instruction allocates for a worst case path to a post-dominator block of said routine.

59. (currently amended) The machine readable medium of claim 54 wherein said one or more functional characteristics include a software pipelined loop[corresponds to a functional characteristic].

60. (currently amended) The machine readable medium of claim 59 wherein [said]the allocation instruction inserted for said software pipelined loop is inserted above said loop in said control flow graph.

61. (original) The machine readable medium of claim 60 wherein said allocation instruction allocates for a worst case path to an exit block of said routine.

62. (original) The machine readable medium of claim 61 wherein said allocation instruction allocates for a worst case path to a post-dominator block of said routine.

63. (currently amended) The machine readable medium of claim 54 wherein said one or more functional characteristics include a function call[corresponds to a functional characteristic].

64. (original) The machine readable medium of claim 54 further comprising sequences of instructions which cause the system to determine the number of registers to be allocated for an allocation instruction after a functional characteristic is found.

65. (currently amended) The machine readable medium of claim 64 wherein all of said one or more functional characteristics within said routine are discovered before said determining is performed.

66. (original) The machine readable medium of claim 64 wherein said determining is performed before a next functional characteristic is discovered.

67. (original) The machine readable medium of claim 54 further comprising sequences of instructions which cause the system to build an understanding of said routine's control flow graph before said searching is performed.

68. (new) A machine readable medium having stored thereon sequences of instructions which are executable by a digital processing system, and which, when executed by the digital processing system, cause the system to perform a method, comprising:

a) performing a first allocation for a first amount of register space at the entry block of a routine;

b) performing a second allocation for a second amount of register space for the live information within said routine at the time of said second allocation;

c) performing a function call to a second routine; and,

d) performing a third allocation for a third amount of register space at the entry block of said second routine, said third amount of register space and said first amount register space having a common register.

69. (new) The machine readable medium of claim 68 wherein said live information is determined by identifying information that is referred to before and after said function call.

70. (new) The machine readable medium of claim 69 wherein said information identified after said function call extends to an exit block of said routine.

71. (new) The machine readable medium of claim 70 wherein the worst case path to said exit block is allocated for.

72. (new) The machine readable medium of claim 69 wherein said information identified after said function call extends to a post-dominator block of said routine.

73. (new) The machine readable medium of claim 72 wherein the worst case path to said post-dominator block is allocated for.

74. (new) The machine readable medium of claim 68 wherein said live information is information that is local to said routine.

75. (new) The machine readable medium of claim 74 wherein a processor said routine is to be executed upon has its associated register space partitioned into register space used only for local information and register space used only for global information, said allocation instruction pertaining only to said register space used for local information.

76. (new) The machine readable medium of claim 68 wherein said live information includes global information.

77. (new) The machine readable medium of claim 68 wherein said second allocation is performed just before said function call.

78. (new) The machine readable medium of claim 68 wherein said second allocation is performed in a pre-dominator basic block of said function call.

79. (new) The machine readable medium of claim 68 wherein said second allocation is performed in said pre-dominator basic block of said function call if there exists a post-dominator basic block of said function call.

80. (new) The machine readable medium of claim 68 further comprising compiling said routine.

COMMENTS

The enclosed is responsive to the Examiner's Office Action mailed on February 13, 2003. At the time the Examiner mailed the Office Action claims 1 through 67 were pending. By way of the present response the Applicant has: 1) added new claims 68 through 80; and, 2) amended claims 9, 14, 15, 16, 19, 20, 23, 25, 28, 35, 42, 49, 54, 55, 56, 59, 60, 63 and 65 for clarification purposes. As such claims 1 through 80 are presently pending. No new matter has been entered. The Applicant respectfully requests reconsideration of the present application and the allowance of claims 1 through 80.

The Applicant brings forward to the attention of the Examiner prior art material that is discussed in matter appearing from page 7, line 21 to page 13, line 8 of the present application's detailed description.

Independent claims 1 and 41

The Examiner has rejected claims 1 and 41 under 35 USC 103 as being obvious in light of U.S. Patent No. 5,950,228 (hereinafter, "Scales") and U.S. Patent No. 5,748,963 (hereinafter, "Orr"). "To establish a *prima facie* case of obviousness ... the prior art reference (or references when combined) must teach or suggest all the claim limitations." MPEP 2143

According to the Examiner's own theory of rejection, it is clear that each and every element of the Applicant's claims 1 and 41 are not found in a combination of Scales and Orr. According to the Examiner, "Scales discloses a

system in which allocation instructions are inserted in a software program" and Orr only teaches "determination of a function call in a code segment" (See, Examiner's Office Action, pg. 2). As such, it is clear that, from the Examiner's own reasoning, the element "if" in the Applicant's claims 1 and 41 can not be found in either Scales or Orr.

On its face, therefore, the Examiner's combination is insufficient to reject independent claims 1 and 41 of the present application. Moreover, given that the Applicant believes that the Applicant is the first to disclose a relationship between the insertion of an allocation routine and the finding of a function call, it is of no surprise to the Applicant that the Examiner's combination fails to find any prior art that covers the word "if" of claims 1 and 41. Therefore, the Applicant respectfully submits that independent claims 1 and 41 of the present application are allowable over the combination of Scales and Orr.

Independent claims 14 and 54

The Examiner has also rejected claims 14 and 54 under 35 USC 103 as being obvious in light of U.S. Patent No. 5,950,228 (hereinafter, "Scales") and U.S. Patent No. 5,748,963 (hereinafter, "Orr"). For clarification purposes the Applicant has amended claims 14 and 54 such that the phrase "if a" has been replaced with the phrase "for each". "To establish a *prima facie* case of obviousness ... the prior art reference (or references when combined) must teach or suggest all the claim limitations." MPEP 2143

Again, according to the Examiner's own theory of rejection, it is clear that each and every element of the Applicant's claims can not be found in a combination of Scales and Orr because no relationship between the insertion of an allocation instruction and the discovery of a functional characteristic can be gleaned from either Scales or Orr. Therefore the element "for each" in the Applicant's claims 14 and 54 is not present in the combination of Scales and Orr. Therefore, the Applicant respectfully submits that independent claims 14 and 54 of the present application are allowable over the combination of Scales and Orr.

Independent Claim 28

The Examiner has also rejected independent claim 28 under 35 USC 103. Section 2141 of the MPEP provides (emphasis added):

> "[w]hen applying 35 USC 103, the following tenets of patent law *must* be adhered to:
> (A) The claimed invention must be considered as a whole;
> (B) The references must be considered as a whole and must suggest the desirability and thus the obviousness of making the combination;
> (C) *The references must be viewed without the use of impermissible hindsight vision afforded by the claimed invention*; and
> (D) Reasonable expectation of success is the standard by which obviousness is determined." MPEP 2141.

With respect to hindsight it is worth noting that "[i]t is impermissible to use the claimed invention as an instruction manual or 'template' to piece together the teachings of the prior art so that the claimed invention is rendered obvious . . . '[o]ne cannot use hindsight reconstruction to pick and choose among isolated

disclosures in the prior art to deprecate the claimed invention.' " In re Fitch 972 F.2d 1260, 1266 (Fed. Cir. 1992).

Here, the Examiner has used a combination of five separate references (Scales, Orr, Proebsting, Wu and Aho) to cover only a modest amount of claim elements that appear in independent claim 28. By so doing, it is clear that the Examiner is using the claimed invention as an instruction manual or template to piece together teachings of the prior art. The Applicant therefore submits that the Examiner's theory of rejection under 35 USC 103 is flawed at least because proper procedural guidelines have not been followed. The Applicant takes this position without making any admission regarding the actual teachings of the references themselves.

As such, the Applicant respectfully submits that independent claim 28 is patentable.

CONCLUSION

In light of the foregoing comments the Applicant respectfully submits that claims 1 through 80 are patentable and the Applicant respectfully requests the allowance of same.

Applicants respectfully submit the present application is in condition for allowance. If the Examiner believes a telephone conference would expedite or assist in the allowance of the present application, the Examiner is invited to call Robert O'Rourke at (408) 720-8300.

Authorization is hereby given to charge our Deposit Account No. 02-2666 for any charges that may be due.

Respectfully submitted,

BLAKELY, SOKOLOFF, TAYLOR & ZAFMAN

Date: 4/14/03

Robert B. O'Rourke
Reg. No. 46,972

12400 Wilshire Boulevard
Seventh Floor
Los Angeles, CA 90025-1026
(408) 720-8300